Demonstrating TabEE: Tabular Embedding Explanations

Roni Copul
Tel Aviv University
ronycopul@mail.tau.ac.il

Nave Frost
eBay reserach
nafrost@ebay.com

Tova Milo
Tel Aviv University
milo@post.tau.ac.il

Kathy Razmadze
Tel Aviv University
kathyr@mail.tau.ac.il

ABSTRACT

We present TabEE, **Tab**ular **E**mbedding **E**xplanations, a framework designed to generate explanations for interpreting tabular embedding models. Our framework aims to furnish both local and global explanations for the original data, facilitating the detection of potential flaws in embedding models. TabEE is versatile and compatible with any tabular embedding algorithm, as it adheres to the black box perspective of embedding models. The generated explanations also enable comparisons between multiple embedding models. This demonstration illustrates the effectiveness of TabEE in providing interpretable insights into tabular embedding models, contributing to improved model understanding and credibility assessment.

PVLDB Reference Format:
Roni Copul, Nave Frost, Tova Milo, and Kathy Razmadze. Demonstrating TabEE: Tabular Embedding Explanations. PVLDB, 17(12): 4285 - 4288, 2024.
doi:10.14778/3685800.3685856

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/KathyRaz/TabEE.

1 INTRODUCTION

Tables serve as vital structures for data organization across a range of domains, from scientific research to business applications. In recent times, the emergence of tabular embedding techniques has introduced a novel approach to analyzing tables and dataframes, offering more intricate data representations. Tabular embedding seeks to capture both structural and semantic aspects of dataframes in a lower-dimensional vector space, enhancing the efficacy of data analysis and processing. In this work, we are focused on the most common type of embedding, where each row or column in a dataframe can be represented as a vector.

Diverse landscape of tabular embedding models. The landscape of tabular embedding techniques has been rapidly advancing, offering diverse options for generating continuous vector representations of tabular data. These methods encompass static and contextualized word embedding methods (i.e. [10]), graph-based approaches (i.e. [3]), and neural network-based methodologies (i.e. [1]), each with distinct strengths. Tabular embeddings have proven valuable in various applications, including table retrieval, entity linking, table completion, and classification tasks, enabling efficient and accurate data analysis ([1, 3]).

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685856

Motivation. While many studies explore tabular embedding techniques, a gap remains in coherently associating between the vector representations and the original attributes. In addition, as models become more complex, understanding and interpreting their mechanisms becomes increasingly difficult. Comprehending these models is crucial for pattern exploration, model comparison, hyperparameter tuning, error analysis and bias identification.

Proposed Solution. Our novel system, Tabular Embeddings Explanations (TabEE), introduces an explanation methodology tailored to dataset-embedding model pairs. By identifying closely grouped data cohorts in the embedding space, we unveil latent patterns within these cohorts by discovering major shifts in distribution across specific original attributes.

Example 1.1. Consider the Cover Type dataset [2], which includes cartographic variables used for predicting forest cover types. In this context, a tabular embedding model is utilized to represent each row of the dataset with a vector, aiming at the classification of the cover type. The primary goal is to interpret patterns encoded within the embedding model and understand their potential implications for tasks, such as error analysis. Figure 1a presents a segment of TabEE's explanation for the provided embedding model and dataset. On the left, clusters are generated using our metric, maximizing aggregated explanation scores, and visualized in a 2-dimensional plot using UMAP [9]. On the right, the distribution of the *Aspect* attribute in the original dataset is illustrated by orange bars, revealing a uniform spread. Conversely, blue bars portray the attribute distribution in the embedding space. Noteworthy is that, for Cluster 0 in the embedding space, corresponding original points predominantly exhibit higher values of the *Aspect* attribute. This observation unveils how the embedding model captures *Aspect* for this cluster, suggesting the formation of a distinctive pattern within this group. Such insights contribute to a deeper understanding of embedding model patterns, indicating that the grouping in the embedding aligns with the values of the *Aspect* attribute.

Demonstration. Our live demonstration presents the functionality of TabEE in generating explanations of real-world tabular datasets and embeddings, as initially detailed in our accepted work for SIGMOD 2024 [4]. TabEE effortlessly integrates with widely-used exploratory data analysis (EDA) platforms, providing users with valuable insights into crucial learned patterns. Demonstrating its compatibility, participants engage with TabEE through Jupyter notebooks, facilitating the exploration of datasets and their selected embedding models. This interactive session allows users to observe explanations, enabling in-depth comparisons between distinct embedding models applied to the same dataset (Section 3).

Related work. One way to improve interpretability is by adapting established techniques from black-box machine learning explanations. While these methods work for model explanations, they often can't fully reveal patterns in tabular embedding models. Existing

methods like TabNet's [1] task-oriented interpretation might not capture the core nature of the underlying embedding model. Also, no prior research systematically compares interpretative abilities of embedding models. Despite connections to related fields like data summarization and exploration tools [5, 12], customized explanations for tabular embeddings are unexplored, highlighting a research gap in addressing interpretability in this context.

2 APPROACH AND METRIC

The intricacies of our algorithm are comprehensively outlined in [4] and briefly summarized here for the purposes of this demonstration.

We consider a dataset represented as a dataframe d, consisting of a list of rows over a schema $\mathscr{A}d$. Each row $r \in d$ represents a tuple, where $A \in \mathscr{A}d$ specifies a dataframe attribute. We denote the dataframe column associated with attribute A as dA, which represents the multiset of all rows' values associated with attribute A in the dataframe d. Given an embedding model M, we denote the representation created for the dataset as Md, also referred to as $\hat{d}$. The embedding representation $\hat{d}$ has a different schema than the original dataset d, such that $\mathscr{A}\hat{d} \neq \mathscr{A}d$. As this work focus on row-level embeddings, each datapoint in the original dataset r has a corresponding embedding representation $\hat{r}$ in the embedding space.

We define a cohort of size n as a set of rows $C = \{r_1, r_2, ..., r_n\} \subseteq d$ that are grouped closely together in the embedding space $\hat{C} = \{\hat{r_1}, \hat{r_2}, ..., \hat{r_n}\} \subseteq \hat{d}$; the set of cohorts partition d.

2.1 Creating Cohorts' Explanations

The vectors generated by the embedding model originate in the embedding space, with no direct connection to the original dataframe. Consequently, it is necessary to establish a connection between these two spaces and quantify the changes occurring within. Inspired by [7], we provide an explanation for a cluster of vectors (cohort), based on their contribution to a pre-defined measure. The Algorithm (fully described in [4]) takes as input an embedding model and the original dataset, and produces a global explanation comprising of several cohort explanations.

The algorithm iterates over values of k from 2 to K_{max} (as explained in the full paper, 10 is a reasonable choice for K_{max}) and performs the following steps for each value of k:

(1) Applies a clustering method, by default K-means, to generate the cohorts, and generates the candidates for the cohort explanation (elaborated in Section 2.2).
(2) Evaluates the quality of each possible combination of candidates (one per cohort), using the metric described in 2.3.
(3) Picks the combination that maximizes the score - this is set to be the explanation for k cohorts.

After generating the explanations for all number of cohorts options, the algorithm selects the number of cohorts that maximizes the score and returns the explanation associated with that number of cohorts. In addition, a 2-dimensional plot of the embedding space is presented, generated using UMAP algorithm [9], in order to create a broader context of the explained cohorts.

2.2 Explaining a Single Cohort

Now, we present the algorithm responsible for generating explanation candidates for the cohorts discovered by the previously described Algorithm. Given a cohort $C = \{r_1 ... r_n\} \subseteq d$, we aim to measure the significance of the change in the distribution of values of a specific attribute A between the original dataset and the cohort C. The deviation, with respect to attribute $A \in \mathscr{A}d$, is defined as the difference between the values in dA and CA, which represent the column associated with attribute A in the original dataframe d and the cohort C, respectively. To quantify this deviation, we utilize the Jensen-Shannon (JS) distance, a well-known statistical test that measures the difference between two distributions, including categorical values. Note that this metric is defined both for numerical and non-cardinal categorical attributes, enbaling a comparison between attributes of different types.

First, we define the column probability distribution $PrdA$ based on the relative frequency of its values. We then calculate the described deviation in distributions, denoted by $DA, C, d := JSPrdA, PrCA$.

Finally, the algorithm selects the m attributes with the largest distances, indicating substantial differences in distribution between the cohort and the original dataset. The algorithm outputs the comparative distributions of attribute values for both the cohort and the original dataset, as illustrated in Figure 1a.

2.3 Metrics for Evaluation

We present a new method for assessing the quality of the explanations generated by TabEE. A complete explanation consists of multiple cohort explanations, each capturing unique model patterns. This metric comprises sufficiency and interestingness metrics for individual cohort explanations and diversity metrics for the overall set of explanations. Later on we prove the connection between the proposed metrics and the ability to capture important patterns of the embedding models, using objective and human-centered experiments.

Sufficiency. To ensure that a given explanation is relevant only to its associated cohort, we define the sufficiency metric, based on the sufficiency metric in [6]. Given an explanation for a cohort, we seek to maximize the number of examples that are assigned with this explanation, from all examples that "correspond" to the distribution of the attribute used in the explanation. Let x be a data tuple from the original dataset d, with its corresponding embedding vector $\hat{x}$. We denote the cohort assigned to $\hat{x}$ as $C_{\hat{x}}$, and the explanation assigned to the cohort as $ExpC_{\hat{x}}$. The explanation is represented as a probability distribution over the values of attribute $A \in \mathscr{A}d$. Then, the probability that the instance x is drawn from the distribution of the explanation e is denoted as the relation Rx, e. This relation quantifies the extent to which the explanation holds for the instance x, taking into account the probability value. Now, the sufficiency of a single example can be formulated as:

$$m^s x = \frac{\sum_{x'} 1\hat{x'} \in C_{\hat{x}} \cdot \sum_{x'} Pr_{x' \sim ExpC_{\hat{x}}} x' \mathrm{Pr} x'}{\sum_{x'} Pr_{x' \sim ExpC_{\hat{x}}} x' \mathrm{Pr} x'}$$

To obtain the global sufficiency, we calculate the local sufficiency for each example and average it across all examples.

Interestingness. This metric gauges the significance of an explanation by comparing attribute distributions in a cohort to the original dataset. For an explanation e_i of cohort C_i, denoting A_i as the attribute, and $PrdA_i$ as its distribution in the original dataset d, interestingness is the distribution difference between PrC_iA_i in the cohort and $PrdA_i$. Using the JS-divergence metric, which accommodates numeric and categorical histograms, this provides a measure of how unique the attribute's cohort distribution is compared to the dataset's. The global interestingness is the average of these measures across all cohorts. It quantifies how intriguing the explanation is in terms of JS-divergence from the overall data distribution.

Diversity. This metric aims to assess the overall quality of multiple explanations for different cohorts, and, quantifies the amount of new information and insights gained from each cohort explanation. It measures the marginal gain of knowledge provided by each new explanation compared to the previously seen explanations. Given a set of explanations $e_1, \ldots, e_k$, where e_i represents the i-th explanation, we define the diversity as the difference between the knowledge gained from e_{k1} and the cumulative knowledge gained from $e_1, \ldots, e_k$. If e_{k1} introduces a new attribute, the diversity is maximal, represented by a value of 1. If it introduces a different distribution for an already known attribute, it contributes the distance between the explanations as a measure of how much of a new knowledge we have gained from the additional explanation. Conversely, if e_{k1} provides the same explanation as a previous one (same attribute and distribution), the diversity is minimal, given value of 0.

To account for the ordering of explanations, we calculate the average diversity value across all permutations of the orderings, considering only explanations that share the same attribute.

Overall score. The overall score combines sufficiency, interestingness, and diversity to comprehensively assess explanation quality. This score is the average of these three components by default, with adjustable weights option. Although the Silhouette score [11] and Dunn index [8] were initially considered, ablation experiments demonstrated their redundancy, leading to their omission.

2.4 External Evaluation

In order to validate the effectiveness of the generated explanations we conducted external experiments, divided into three parts: **(1) Objective Evaluation**: We employ automated methods to analyze default system parameters, using ablation experiments; assess the trade-off between running time and explanation quality; evaluate general alignment of the algorithm with the patterns of the underlying embedding model; and evaluate the relevance of the patterns unraveled by TabEE to downstream tasks, using a comparison to the explanations generated by common black box explanation methods, such as SHAP. **(2) Baselines Comparison:** We explored alternative methods for creating explanations of the embedding space, instead of using distributions shifts, such as Association Rules and cluster representatives. Additionally, our method is compared to other explainability algorithms, designed for interpreting downstream tasks. **(3) Human-Centered Experiments:** We conducted a user study to validate our metric for measuring explanation quality, by assessing the increase in user productivity when following machine-produced explanations versus using traditional data analysis. Additionaly, we analyze the correlation between our method's scores and a subjective preference of the users.

System parameters optimization. The following system parameters were evaluated and optimized through ablation experiments: *(1) Number of candidates per cluster*: The default is three, balancing quality and efficiency. Increasing candidates improves quality but increases running time. *(2) Range of clusters examined*: The default range is 2 to 10. *(3) Number of explanations per cluster*: By default, one explanation is output per cluster in the global explanation. *(4) Importance of metric components*: by default equal weights are assigned. *(5) Distributions distance metric*: Jensen-Shannon (JS) metric is used. *(6) Binning methods*: We use FEDEX's [7] binning method for discretizing numeric attributes, ensuring equal distribution within each bin for better comparison and distance calculation. *(7) Clustering method*: The default is KMeans, chosen for it's simplicity and fast runtime.

Although default values are set for these system parameters, they can be altered by the users, in order to fit their specific needs.

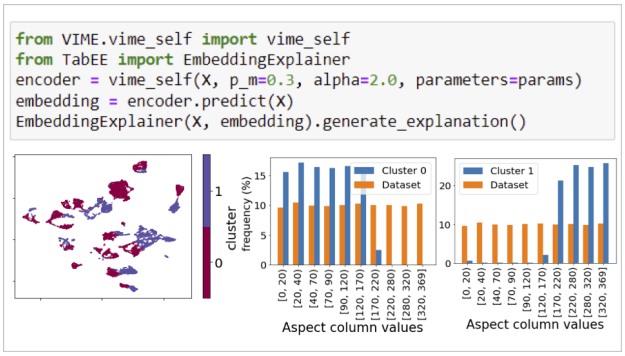
Synopsis of Findings. The comprehensive outcomes of our experiments can be found in [4]. Our results underscore the efficacy of TabEE in generating explanations that adeptly capture the learned patterns of the embedding model. Notably, a robust correlation exists between elevated scores from our metric and the actual quality of explanations, as corroborated by both objective empirical experiments and subjective assessments from our user study. Every facet of our proposed metric, including the distance metric for distributions and configuration parameters such as the maximum number of scrutinized clusters, has been affirmed as essential and precise. Additionally, our user study attests that users employing our system exhibited enhanced objective performance, such as the proficiency to discern data patterns and comprehend distinctions among diverse embedding models for subsequent analysis.

The goal of this demo is to affirm the practicality and efficacy of TabEE in augmenting data analysis and facilitating model comprehension, through live demonstration in practical scenarios using classical EDA setting.

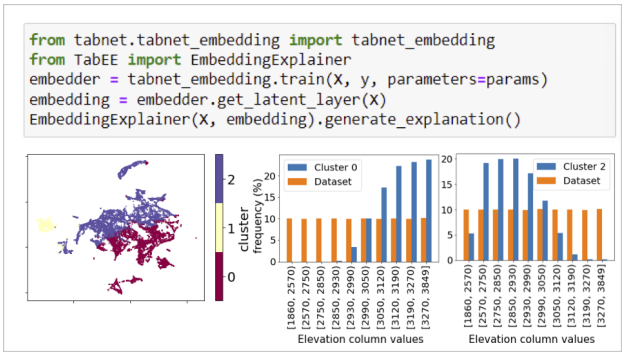
3 SYSTEM AND DEMO OVERVIEW

We demonstrate TabEE, a user-friendly tool for exploring table embedding models in EDA through a common scenario for classification of Cover Type dataset [2]. TabEE integrates pre-trained embedding models into EDA workflows, such as Jupyter notebooks, using a local Python 3.8 library. empower users with a detailed analysis of the model's performance and patterns. It comes with default settings, detailed in Section 2.4. The following sections provide an overview of the demonstration scenarios, starting with an introduction to the system and its usage, followed by a comparison of two embedding models. Note that the scenarios depicted, including Figure 1, are based on a simplified setup with two embedding models for the Cover Type dataset, featuring 2-3 cohorts and a single explanation for each. For other datasets and models, the explanations may be more involved.

Scenario 1: Initial model exploration with TabEE. Our demonstration begins with an interactive exploration of TabEE. Participants delve into a tabular embedding model, such as TabNet [1], trained on



(a) VIME embedding example



(b) TabNet embedding example

Figure 1: Comparison of two embedding methods applied on Cover Type dataset, generated using TabEE

the Cover Type dataset, used for tasks such as predicting forest cover types (precalculated for time efficiency). This immersive experience helps users understand the fundamental patterns uncovered by the embedding model. For instance, as shown in Figure 1b, TabEE generates explanations featuring 3 cohorts. These explanations reveal that the embedding model strongly emphasizes on the *Elevation* column, as both of the two largest cohorts capture it prominently, within different ranges of values. In contrast, the third cluster is smaller in size, therefore capturing fewer tuples and being less significant for the embedding model.

Scenario 2: Comparative analysis with TabEE. In the second scenario, participants select an alternative tabular embedding model from a curated collection, including VIME [13] and EmbDI [3]. Guided by expert assistance and TabEE's explanations, attendees compare the suitability of a new model for a downstream task against the previously trained model. Participants offer insights into the performance differences and suggest improvements. Additional downstream tasks and models are also available for exploration.

To illustrate, both selected embedding models are trained on the Cover Type dataset. Subsequently, an XGBoost model is trained separately on these representations, and its performance is evaluated for each of the models. By examining the explanations generated by TabEE for both models, as depicted in Figure 1, participants gain insights into why one embedding model (TabNet) outperforms the other (VIME) on this downstream task. For example, the VIME embedding is explained by two cohorts that mainly focus on the *Aspect* column capturing two different ranges of values for this attribute. In contrast, the explanation for the TabNet embedding highlights the significance of the *Elevation* attribute. Analyzing the correlations between Aspect and Elevation attributes to the label reveals a substantial gap, with the *Elevation* attribute showing a significantly higher correlation with the label. This difference in correlation likely explains the performance disparity between the models. Although simplistic, this comparison allows us to analyse the patterns captured by the embedding in a higher level, and as a follow-up to dig deeper into sub-patterns of the large clusters.

Finally, users can explore the system's inner workings, adjusting parameters to see how preferences affect generated explanations. As detailed in [4], some users might prefer higher diversity values for broader analysis, while others might prioritize sufficiency for more precise explanations. Additionally, users get the chance to tune the number of explanations per cluster, the clustering method, and the other system parameters detailed in Section 2.4, in order to understand their possible impact of the generated explanations.

ACKNOWLEDGMENTS

The research was partly supported by ISF - the Israel Science foundation - grant 2707/22 of the Breakthrough Research Grant (BRG) Program.

REFERENCES

[1] S. Ö. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6679–6687, 2021.
[2] J. Blackard. Covertype. UCI Machine Learning Repository, 1998.
[3] R. Cappuzzo, P. Papotti, and S. Thirumuruganathan. Embdi: generating embeddings for relational data integration. In *29th Italian Symposium on Advanced Database Systems (SEDB), Pizzo Calabro, Italy*, 2021.
[4] R. Copul, N. Frost, T. Milo, and K. Razmadze. Tabee: Tabular embeddings explanations. In *Proceedings of the 2024 ACM SIGMOD*, 2024.
[5] J. P. Cunningham and Z. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *J. Mach. Learn. Res.*, 16(1), 2015.
[6] S. Dasgupta, N. Frost, and M. Moshkovitz. Framework for evaluating faithfulness of local explanations. In *International Conference on Machine Learning*, pages 4794–4815. PMLR, 2022.
[7] D. Deutch, A. Gilad, T. Milo, A. Mualem, and A. Somech. Fedex: An explainability framework for data exploration steps. *VLDB*, 15(13):3854–3868, 2022.
[8] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.
[9] L. McInnes, J. Healy, N. Saul, and L. Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29), 2018.
[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013.
[11] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.
[12] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics*, 25(1):672–681, 2018.
[13] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33:11033–11043, 2020.